WHAT IS CLAIMED IS:

1. A method of symmetric multiprocessing in which one or more processors, a first memory medium storing a micro kernel operating system in a machine executable form and a second memory storing a thread scheduler in a machine executable form are interconnected via a communication network, said method comprising the steps within said thread scheduler of:

responding to a thread requiring a call to said micro kernel operating system by requesting a global lock;

responding to said global lock being available by performing the steps of:

acquiring said global lock from said thread scheduler;

performing said call to said micro kernel operating system; and

releasing said global lock.

2. A method of symmetric multiprocessing as claimed in claim 1, wherein said micro kernel operating system comprises a pre-emptable micro kernel operating system, said method further comprising the steps with said thread scheduler of:

pre-empting any non-critical threads currently executing on said pre-emptable micro kernel operating system prior to said step of executing said thread on said pre-emptable micro kernel operating system; and

reinstating said pre-empted threads following said step of executing said thread on said pre-emptable micro kernel operating system.

3. A method of symmetric multiprocessing as claimed in claim 2 wherein said step of performing said call to said micro kernel operating system comprises the steps of:

entering said pre-emptable micro kernel operating system;

executing operating system functions as required by said thread;

locking said pre-emptable micro kernel operating system; and

exiting said pre-emptable micro kernel operating system.

4. A method of symmetric multiprocessing in which one or more processors, a first memory medium storing a pre-emptable micro kernel operating system in a machine executable form and a second memory storing a thread scheduler in a machine executable form are interconnected via a communication network, said method comprising the steps within said thread scheduler of:

responding to a thread requiring a call to said micro kernel operating system by requesting a global lock;

responding to said global lock being available by performing the steps of:

pre-empting any non-critical threads currently executing on said pre-emptable micro kernel operating system;

acquiring said global lock from said thread scheduler;

entering said pre-emptable micro kernel operating system;

executing operating system functions as required by said thread;

locking said pre-emptable micro kernel operating system;

exiting said pre-emptable micro kernel operating system;

releasing said global lock; and

reinstating said pre-empted threads.

5. A computer system comprising;

one or more processors;

a first memory medium storing a pre-emptable operating system in a machine executable form;

a second memory storing a lock manager in an machine executable form;

a communication network interconnecting said one or more processors, said first memory and said second memory; and

said lock manager being operable to:

responding to a thread requiring access to a critical area of said pre-emptable operating system by requesting a global lock;

responding to said global lock being available by performing the steps of:

pre-empting any non-critical threads currently executing on said operating system;

executing said critical thread on said operating system; and

reinstating said pre-empted threads.

add A1

Add C7